

OBFUCIA | Documentation

Project Plan

Contents

Scope Statement 3

- Project Justification 3
- High-Level Requirements 3
- Limits and Exclusions 3
- Summary of Project Deliverables 4
 - Project Management-Related Deliverables 4
 - Product Related Deliverables 4
- Project Success Criteria 4

Organizational Chart 5

Gantt Chart 6

Tools and Standards 7

- Development Environment 7
- Process Model 8
- Coding Standards 9
- Documentation Standards 9

Configuration Management Plan 10

Weekly Status Reports 10

Figures, Tables, and Listings

Figures

Figure 1.0 – Organizational Chart 5

Figure 1.1 – Gantt Chart 6

Figure 1.2 – JIRA Screenshot 8

Figure 1.3 – GitLab Screenshot 9

Scope Statement

Project Justification

Roger West, our boss, has requested the development of this project to help a client develop a C# internal standalone application that allows text and image file types to be hidden in a targeted media files. This will allow others with the proper security clearance to discreetly obfuscate and send sensitive media without being detected.

High-Level Requirements

- OBFUCIA will run on a Windows 10 computer using C#.
- OBFUCIA will run using UWP (Universal Windows Platform).
- End users should be able to upload text and image media from the local computer or be able to take picture of images using a webcam.
- Allow the option of an image be hidden in a target image file.
- Allow the option of ASCII text be hidden in a target image file.
- An option to save processed media to a users local machine.

Limits and Exclusions

Network functionality should be limited to functionality to the database, cloud, or FTP server. End users will not be provided a webcam and must provide their own.

The following features can be added ONLY if time permits:

- Hiding ASCII text in a target video file.
- Hiding ASCII in a target audio file.
- Hiding audio in a target image file.
- Additional security such as permissions to data to certain users.

Summary of Project Deliverables

Project Management-Related Deliverables

- Project Scope Statement
- Work Break Down Schedule
- GANTT Chart
- Organizational Chart
- Communication Plan
- Coding Standards
- Primary Project Plan (this entire document)
- Software Process and Version Control

Product Related Deliverables

Software: Software will be able to support files that contain ASCII, webcam support, and any other software necessary to run the program.

User Documentation: Instructions for the end users on how to install and use the OBFUCIA software.

Programming Manual: A programming archive will be created and maintained for future development of OBFUCIA.

Project Success Criteria

Project success will be measured by timely deliverables as per directed by the guidelines for the capstone course.

Organizational Chart

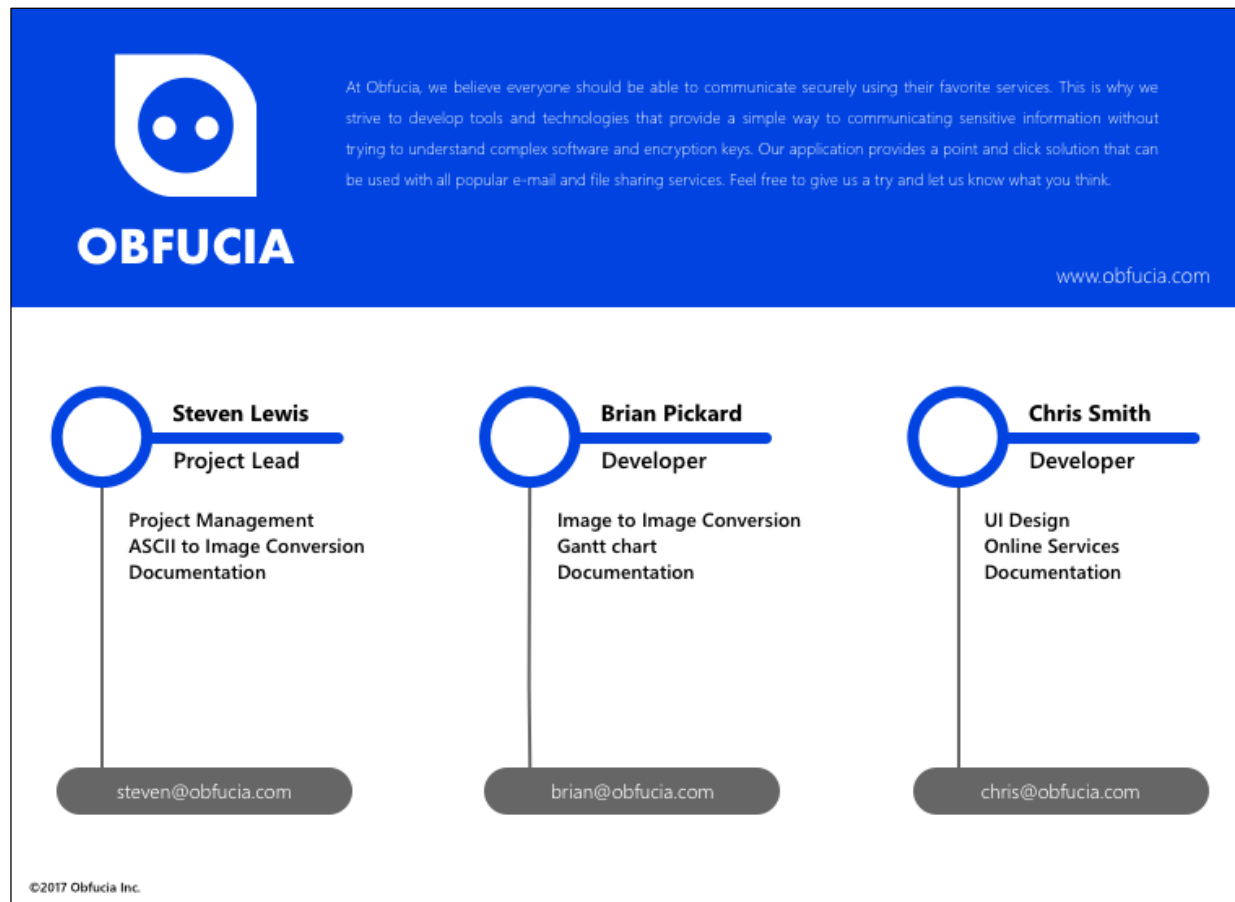


Figure 1.0 – Organization Chart

Organization Chart file can be downloaded at the following URL:
<https://www.obfucia.com/docs/orgchart.pdf>

Gantt Chart

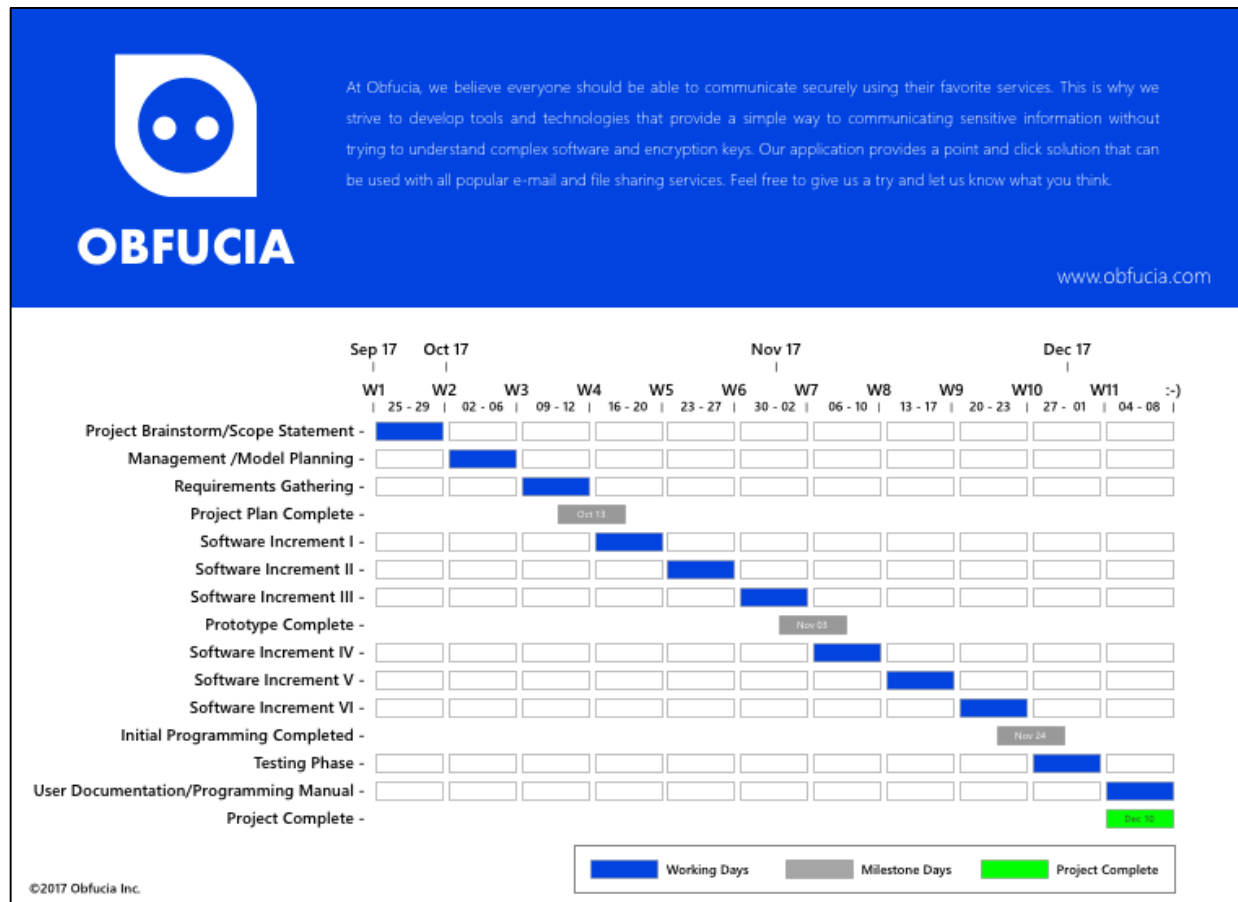


Figure 1.1 – Gantt Chart

Organization Chart file can be downloaded at the following URL:
<https://www.obfucia.com/docs/ganttchart.pdf>

Tools and Standards

Development Environment

Development environments should be standardized as much as possible to make troubleshooting easier and to reduce environmental variables as a cause for application errors. The following development standards should be adhered to for all developers working on the OBFUCIA application:

Hardware

Platform:PC | Proc:x86-64 | RAM:8GB+ | HD:50+ | Res:1024x640

Operating System

Microsoft Windows 10 Fall Creators Update (10.0; Build 16299)

Development Tools

Integrated Development Environment

Microsoft Visual Studio 2017 Enterprise (Version 15.3.5)

Version Control

Git for Microsoft Windows (Version 2.15.1)

Design

Blend for Microsoft Visual Studio Enterprise

Fonts

Microsoft Segoe UI | Futura

Process Model

Obfucia development will utilize an iterative process model, specifically the Agile development methodology to allow for maximum flexibility and rapid development. To manage this process, we will use Atlassian's JIRA and GitLab as tools to manage the process.

JIRA provides a streamlined environment to manage agile sprints:

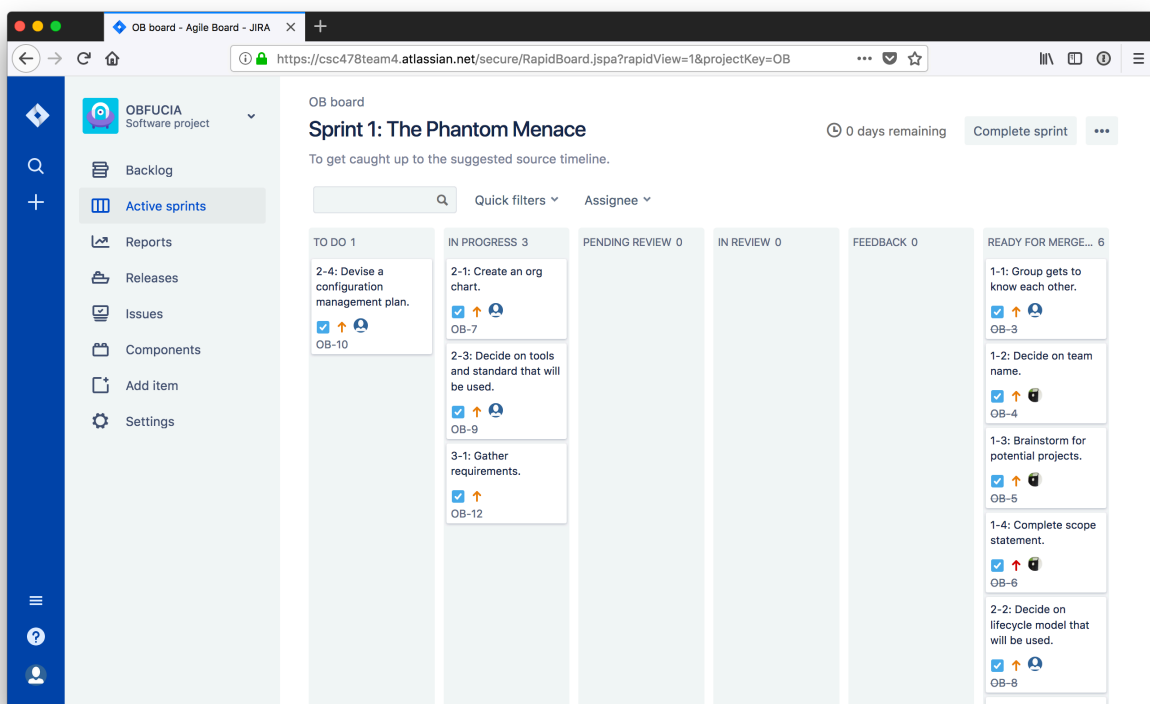


Figure 1.2 – JIRA Screenshot

All of the projects tasks are loaded into the backlog and assigned to a “sprint” every week. Then, developers will assign themselves tasks to complete to meet the weeks sprint requirements, and the project leader will then review the sprint once it's complete. This process will repeat until the application is shipped.

Another important tool in managing the process model is GitLab:

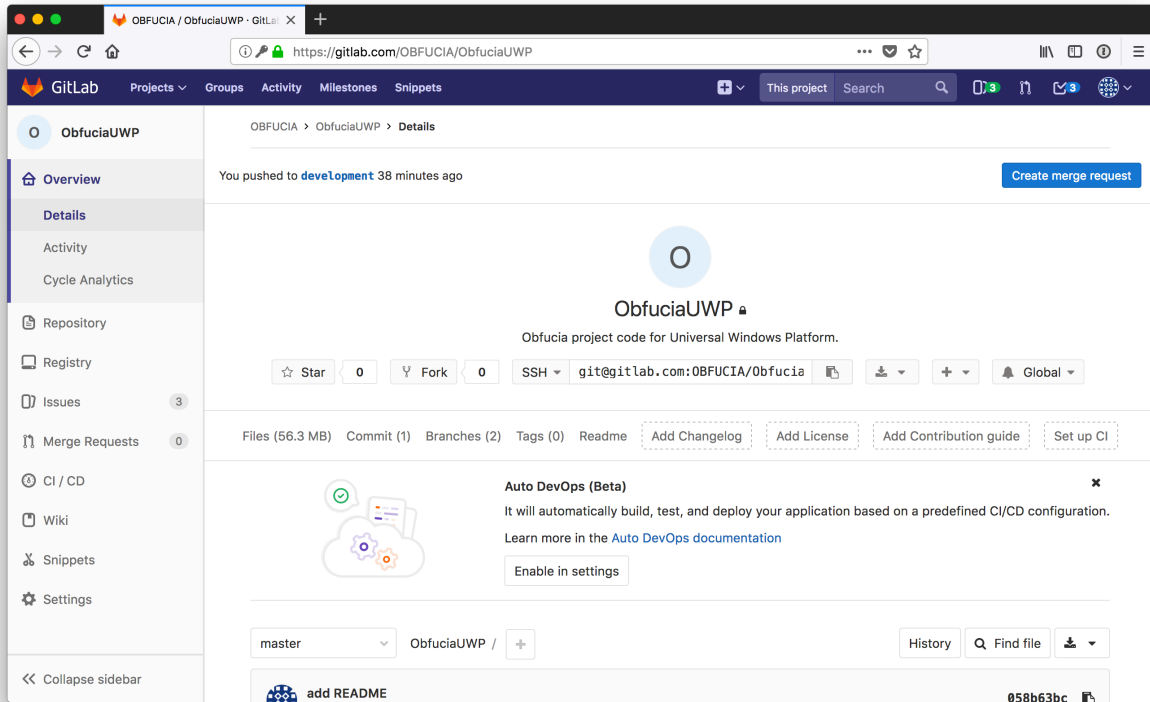


Figure 1.3 - GitLab Screenshot

GitLab is used both as a repository for active code using git, and as a source of developer documentation using the wiki.

Coding Standards

The coding standards for this project are very simple and require only commenting all significant sections of code, functions, and other areas. Some of the components are standard Microsoft templates provided as part of the UWP libraries. These will be indicated as part of the code comments.

Documentation Standards

All documentation will meet a standard look, feel, and format. Documentation will be provided in Adobe PDF format available as part of the final package and online.

Configuration Management Plan

All configuration management and version control will be handled by git using GitLab. This will serve two purposes: one, all project members can work on the same code simultaneously, and two, we have an automated means of tracking changes without manually versioning files, folders, and other components. Using GitLab also allows us to bring new project members onboard and integrate them into our team's workflow quickly. Git allows us to roll-back the version we are working on, and ignore changes should active development cause issues.

There will be two git branches used during development, development and master. The development branch will hold the code base that is in active development. The Master branch will hold the code base that is currently approved for deployment.

Versioning of the releasable application will use a #.# format. The first number is a major release version, the second number is an incremental update number. The first version of Obfucia will be 1.0.

Weekly Status Reports

All weekly status reports will be provided to Roger West at rwest2@uis.edu by our project leader.